

Scalar Linear Network Coding for Networks with Two Sources

Sormeh Shadbakht
Electrical Engineering Department
California Institute of Technology
Pasadena, CA 91125
Email: sormeh@caltech.edu

Amin Jafarian
ECE Department
University of Texas at Austin
Austin, TX 78712
Email: jafarian@ece.utexas.edu

Babak Hassibi
Electrical Engineering Department
California Institute of Technology
Pasadena, CA 91125
Email: hassibi@caltech.edu

Abstract—Determining the capacity of networks has been a long-standing issue of interest in the literature. Although for multi-source multi-sink networks it is known that using network coding is advantageous over traditional routing, finding the best coding strategy is not trivial in general. Among different classes of codes that could be potentially used in a network, linear codes due to their simplicity are of particular interest. Although linear codes are proven to be sub-optimal in general, in some cases such as the multicast scenario they achieve the cut-set bound. Since determining the capacity of a network is closely related to the characterization of the entropy region of all its random variables, if one is interested in finding the best linear solution for a network, one should find the region of all linear representable entropy vectors of that network. With this approach, we study the scalar linear solutions over arbitrary network problems with two sources. We explicitly calculate this region for small number of variables and suggest a method for larger networks through finding the best scalar linear solution to a storage problem as an example of practical interest.

I. INTRODUCTION

There recently has been a great deal of effort to determine the information-theoretic capacity of networks. However for long, the basis of the main technique for sending information over networks was to consider information as a fluid which could only be routed (or replicated). Network coding, first introduced in [1], showed that for networks with two destinations or more, routing is not optimal and coding at the nodes of the network can in general increase the throughput and save the bandwidth. Nonetheless the optimal coding strategy remains as a topic of research.

In the multicast scenario, where all the destinations desire the same set of source messages, linear network coding is proven to achieve the cut-set bound. For the general multi-source multi-sink networks where sinks can have arbitrary demands, the capacity region is expressed in terms of the space of *entropy vectors*—the entropy region—of random variables associated with the network [2][3]. This characterization yields the best rates possible, independent of the coding used to achieve them.

Definition 1 (Entropy Region). Let X_1, \dots, X_n be n random variables. Define $h(X_\alpha) = h(X_i, i \in \alpha)$ as the joint entropy of the random variables indexed by the elements of the set $\alpha \subseteq \mathcal{N} = \{1 \dots n\}$. The $2^n - 1$ dimensional vector of all these

joint entropies together yields an entropy vector. Conversely any $2^n - 1$ dimensional vector whose entries can be considered as the joint entropies of a collection of n random variables is called *entropic*. The space of all $2^n - 1$ dimensional entropy vectors is called the *entropy region* and denoted by Γ_n^* [4].

Since the characterization of Γ_n^* is an open problem for more than 3 variables, explicit computation of the capacity region remains unsolved. However one might be interested in obtaining the capacity region of the network when using a specific group of codes.

Definition 2 (Linear Code). Let $G = (V, E)$ denote a network graph with the node set V and the edge set E . Denote the variables associated with the sources and the edges of the network by X_k which are assumed to be vectors of length m over some finite field \mathcal{A} . Let $X_{in}(i)$ and $X_{out}(i)$ be the input and output variables for node V_i respectively, where $X_{in}(i)$ includes source messages when i is a source and $X_{out}(i)$ includes demands when i is a sink. We call a coding on the network linear if each output variable is obtained from a linear combination of the inputs, i.e. for any $X_k \in X_{out}(i)$,

$$X_k = \sum_{j: X_j \in X_{in}(i)} F_{jk} X_j \quad (1)$$

where F_{jk} are matrices with elements over \mathcal{A} . When $m = 1$ the code is called scalar linear as opposed to vector linear [5].

It is shown that every solvable multicast network has a scalar linear solution over a sufficiently large alphabet size [6][7]. Although later it was proved that for the case of non-multicast networks even vector linear coding is sub-optimal [5], linear codes are of particular interest due to their simplicity.

In this paper we will focus on the best scalar linear codes for small arbitrary networks, especially those with two sources. In Section II we will state some known results about how the scalar linear coding capacity is related to the region of scalar linear representable entropy vectors. In Section III we give an algorithm and explicitly compute the scalar linear representable entropy region for 4 random variables. The method is extendable to greater number of random variables as well. We also state that to find the best scalar linear solution to a network with given number of sources only a subset of

the whole scalar representable entropy region will be needed in practice. We then present a method to compute it for small networks with 2 sources. Finally in Section IV we find the best scalar linear solution to a storage problem and show how our techniques can be used in larger networks where the methods of Section III become computationally prohibitive.

II. PRELIMINARIES

As stated in the introduction, the linear coding capacity of a network hinges on the region of linear representable vectors.

Definition 3 (Linear representable vectors). [8] A $2^n - 1$ dimensional vector g whose entries are indexed by subsets of $\mathcal{N} = \{1, \dots, n\}$, is called *linearly representable*, if there exist n vectors $\{v_1, \dots, v_n\}$ ¹ over a finite field $GF(q)$ such that for any $\alpha \subseteq \mathcal{N}$,

$$g_\alpha = \log q \cdot \text{rank}(\oplus_{i \in \alpha} v_i) \quad (2)$$

where $\oplus_{i \in \alpha} v_i$ denotes the space spanned by $\{v_i, i \in \alpha\}$.

A linearly representable vector is entropic.

Lemma 1 (Linear representables are entropic). Let the vectors $\{v_1, \dots, v_n\}$ over the finite field $GF(q)$ provide a linear representation for g such that $g_\alpha = \log q \cdot \text{rank}(\oplus_{i \in \alpha} v_i)$. Let $M = [v_1^T \mid \dots \mid v_n^T]^T$ and $a = [a_1, \dots, a_p]^T$. Define $X = [X_1, \dots, X_n]^T$ as,

$$\begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} \quad (3)$$

where each a_i is uniformly distributed over the finite field $GF(q)$. Then the joint entropies of X_i 's will be,

$$h(X_\alpha) = \log q \cdot \text{rank}(v_\alpha) \quad (4)$$

where v_α is a submatrix obtained by taking rows v_i where $i \in \alpha$. Therefore $g_\alpha = h(X_\alpha)$ [9].

Note that in Lemma 1, if v_i 's are $1 \times p$ vectors we say that the vector g is *scalar linear representable*, while if they are $m \times p$ for some integer $m > 1$, each random variable X_i is also m dimensional and we say that the set of v_i compose a *vector linear representation* for g .

Lemma 1 shows that the region of linearly representable vectors is a subset of Γ_n^* . Denote the space of $2^n - 1$ scalar linear representable vectors over the finite field $GF(q)$ by Φ_n^* .

Theorem 1 (Linear representability and scalar linear network codes). Let X_1, \dots, X_n denote the variables of a wired network. The best achievable rate by scalar linear codes can be obtained from the following optimization over the convex

cone of scalar linear representable entropy vectors,

$$\begin{aligned} & \max h(X_S) \\ & \text{s.t. } h(X_S) = \sum_{i \in S} h(X_i) \\ & \quad h(X_k | X_{in}(i)) = 0 \quad X_k \in X_{out}(i) \\ & \quad h(X_i) \leq c_i \quad i \notin S \cup T \\ & \quad h \in \text{con}(\Phi_n^*) \end{aligned} \quad (5)$$

where $\text{con}(\cdot)$ represents the convex cone and S and T refer to the set of source and sink variables respectively. Each c_i denotes capacity of the link over which X_i is being sent.

It is known that the linear representable entropies satisfy an inequality called the *Ingletton inequality* [10] which does not hold for all entropy vectors. While the Ingletton inequality provides an inner bound for Γ_n^* it also proves that the region of all linear representable entropies is a strict subset of Γ_n^* . This justifies the sub-optimality of linear codes. It is worth mentioning that the full characterization of (scalar or vector) linear representable entropies for any number of variables remains open.

For 4 random variables it has been shown in [11] that the region of linearly representable entropies is completely characterized by the set of Ingletton and Shannon-type inequalities. The approach of [11] is based on finding all the extreme rays of the set defined by Ingletton and Shannon-type inequalities and finding a linear representation for all those points. Although this approach finds the whole linear representable region for 4 variables it is not extendable to more than 4. Moreover it does not find the region of *scalar* representable entropies.

In an attempt to find the capacity region of wired networks based on Theorem 1, we will henceforth focus on Φ_n^* .

III. COMPUTING THE REGION OF SCALAR LINEAR REPRESENTABLE ENTROPY VECTORS

Here we will obtain the region of scalar representable vectors for 4 random variables in a systematic fashion which is extendable to more variables. Moreover we will show that the same framework can be used to obtain the entropy region of all feasible entropy vectors in small networks with two sources.

A. 4×4 Matrix

Based on Lemma 1 one can obtain the set of all linear representable vectors g for n random variables by finding all the possible rank vectors of the matrix $M = [v_1^T \mid \dots \mid v_n^T]^T$ where $(\cdot)^T$ denotes the transpose operation.

Theorem 2 (Region of linear representable entropies). Φ_n^* is obtained from all the rank functions of an $n \times n$ matrix M with entries over an arbitrary finite field $GF(q)$. The matrix M can be assumed to be lower triangular without loss of generality.

Proof: First part follows trivially from Lemma 1 and the latter part is established by repeated use of the Bezout identity. ■

Therefore in order to compute Φ_4^* we need to find all rank vectors obtained from a lower triangular 4×4 matrix M for

¹Note that $\{v_1, \dots, v_n\}$ could as well be subspaces of a vector space V over $GF(q)$

which we need to consider all relative dependencies between rows of M . The following matrix shows a general structure:

$$M = \begin{bmatrix} x & & & \\ x & x & & \\ x & x & x & \\ x & x & x & x \end{bmatrix} \quad (6)$$

Note that since the matrix is lower triangular, in calculating the ranks of a collection of rows only the position of the last non zero elements of rows become important. Let M_i be the i 'th row of M and p_i be the position of the last nonzero entry of M_i . Now we can describe the algorithm for computing the rank vector from a lower-triangular 4×4 matrix.

1) *Algorithm for computing the rank vectors from a 4×4 matrix:* The general idea is to go over all possible structures for the matrix M by considering all position vectors $P = [p_1 \ p_2 \ p_3 \ p_4]$ where $p_i = 0, \dots, i$. Then for each P , we determine all the dependency relations between the rows of M . This is done in two steps, first pairwise dependencies are assumed and then based on that, triple dependencies are investigated. For many values of P there are more than one possible dependency relations between rows. Once the dependency relations are set, the rank vector can be obtained easily [12]. Note that the underlying field size is arbitrary. At the end, rays of the convex cone of all the so-obtained rank vectors are determined via use of the Fourier-Motzkin elimination through a program. Details are omitted for lack of space however the numerical results are given in the next subsection. The following example illustrates the process.

Example: Let M be of the following form,

$$M = \begin{bmatrix} 0 & & & \\ x & x & & \\ x & x & & \\ x & x & & \end{bmatrix} \quad (7)$$

Therefore we have, $p_1 = 0$, $p_2 = p_3 = p_4 = 2$. Note that $p_1 = 0$ means that the entire row M_1 is zero. The rank vector of this matrix can be obtained as follows. First we compute the ranks of single rows, $r_1 = 0$, $r_2 = r_3 = r_4 = 1$ where r_α will be used to denote the rank of the collection of rows indexed by α . Among the pairs, r_{1i} are also easy to compute, in fact, $r_{1i} = 1$ $i = 2, 3, 4$. However special care should be given to the computation of the rest of the pairwise ranks. Since $p_2 = p_3 = p_4$, rows 2,3,4 can be dependent or independent with respect to each other. However their dependencies are tied together and one needs to consider the dependency relation of all rows ending in the same position with respect to each other and then compute the rank of their subsets. For example one cannot consider that rows 2 and 3 and also 2 and 4 be dependent but 3 and 4 be independent. We conclude that in this case there can be 3 different scenarios,

C1) rows 2 and 3 and 4 are all independent from each other, $r_{23} = r_{24} = r_{34} = 2$, $r_{ijk} = 2$, $r_{1234} = 2$

C2) two of the three rows i and j are linearly dependent but are independent from the third row k , $r_{ij} = 1$, $r_{ik} = r_{jk} = 2$, $r_{1ij} = 1$, $r_{1ik} = r_{1jk} = r_{ijk} = 2$, $r_{1234} = 2$

TABLE I
RAYS OF THE SCALAR LINEARLY REPRESENTABLE ENTROPIES

(1)	(1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3)
(2)	(1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)
(3)	(1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2)
(4)	(1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2)
(5)	(1, 1, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2)
(6)	(1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2)
(7)	(1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2)
(8)	(1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2)
(9)	(0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1)
(10)	(0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1)
(11)	(0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2)
(12)	(0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1)
(13)	(1, 0, 1, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2)
(14)	(1, 1, 0, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2)
(15)	(1, 1, 1, 0, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2)
(16)	(1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0)
(17)	(0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1)
(18)	(0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1)
(19)	(0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1)
(20)	(1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1)
(21)	(1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1)
(22)	(1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1)
(23)	(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
(24)	(1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
(25)	(1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
(26)	(1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
(27)	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)

C3) all pairs are dependent, $r_{23} = r_{24} = r_{34} = 1$, $r_{123} = r_{124} = r_{134} = r_{234} = r_{1234} = 1$

2) *Region of Scalar Linearly Representable Entropies for 4 Random Variables:* The total number of such obtained rank vectors are 162 out of which 68 are the corner points of their convex hull which is as well represented by 86 inequalities or equivalently hyperplanes. Among the hyperplanes 50 of them pass through the origin and are used for computing the rays of the convex cone of the rank vectors. Rays of the convex cone include 27 vectors which are stated in Table (I).

Comparisons: In [13] the corner points of the entropy vectors of 4 quasi-uniform random variables with alphabet size 2 are obtained. Interestingly there are 67 such corner points and they all belong to the set of 68 corner points of the convex hull of the scalar rank vectors that we have obtained here. The only differentiating vector is, (1 1 1 1 2 2 2 2 2 2 2 2 2 2) which is not representable over binary alphabet size. Therefore all the other corner points of scalar ranks correspond to the entropies of quasi-uniform random variables with binary alphabet size.

Another comparison with the results of [11] shows that the rays of the scalar rank region are exactly the scalar representable rays of the vector rank region. In fact [11] finds 35 rays for the set defined by Shannon type and Ingleton inequalities and shows that they are all representable. However only 27 of those are scalar representable which exactly correspond to the rays of the convex cone region that we have found.

B. $n \times 2$ matrix

As stated in Theorem 1, the region of linearly representable entropy vectors is important in finding capacity region of a given network. In the last section we found this region for 4

random variables. However most networks involve more than 4 variables and finding the linear representable entropy region becomes computationally hard when the number of variables grow. Therefore in an attempt to find linear solutions for a general network, one may simplify the problem by limiting the number of sources and therefore only considering joint entropies bounded by the number of sources.

Theorem 3. *The entropy region of all scalar linear solutions of a network with two sources and n variables is a subset of the convex cone of all scalar rank vectors of an $n \times 2$ matrix.*

Proof: Follows from Lemma 1 and that there are only two independent variables, (i.e. source messages) propagated through the network. ■

Denote the space of scalar rank vectors obtained from an $n \times 2$ matrix by $\Phi_{n,2}^*$.

Theorem 4. *Given a network with two sources, a scalar linear solution for it can be found by solving a linear program over the convex cone $\Phi_{n,2}^*$.*

Proof: Follows from Theorems 3 and 1. ■

Henceforth we will try to find the convex cone of the rank vectors of an $n \times 2$ matrix.

1) *General Algorithm:* The algorithm for finding all the rank vectors of an $n \times 2$ follows closely that of a 4×4 matrix described in the last section. Here we will just give the general outline. In contrast to the 4×4 case, there is no lower triangular matrix and in considering different structures for the $n \times 2$ matrix, N , we only need to consider that whether each row is zero or not zero. The following shows a general structure for N when $n = 4$,

$$N = \begin{bmatrix} x & x \\ x & x \\ 0 & 0 \\ x & x \end{bmatrix} \quad (8)$$

Similar to the 4×4 case, we first determine the pairwise dependencies between rows (Lemma 2). This tells us which rows are aligned to each other and enables us to determine all the pairwise ranks. Next since N has rank at most two, all the other entries of the rank vector can be simply computed from the pairs (Lemma 3).

Lemma 2. *Let $K = \{i_1, \dots, i_k\}$ denote the set of nonzero rows of an $n \times 2$ matrix N . Then there is a bijection between all rank vectors of N and all set partitions of K .*

Lemma 3. *All the entries of the rank vector which correspond to ranks of a set of more than 2 rows, can be obtained from the pairwise ranks.*

2) *Some Explicit Computations:* Now we give the results of the explicit computation in three cases.

- 1) 4×2 matrix: Computing all the ranks of a 4×2 matrix results in 52 vectors. The convex hull of these vectors is represented by 156 inequalities and interestingly all the 52 points are corner points of the

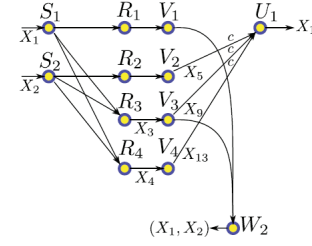


Fig. 1. Network Model

convex hull. Out of 156 inequalities, 50 of them are homogenous, meaning that they define hyperplanes that pass through the origin. However as opposed to the 4×4 case there are only 26 rays for the convex cone. The only missing vector compared to the 4×4 case is, $(1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3)$. This is somehow not surprising, since this is the only ray in Table I that has rank entries greater than 2.

- 2) 5×2 matrix: There are a total of 203 vectors, out of which 112 are rays of the convex cone region.
- 3) 6×2 matrix: There are 877 rank vectors for a 6×2 matrix which are obviously 63 dimensional. Computing the convex cone of these vectors by means of a linear program, gave 575 rays.

IV. SOLVING A NETWORK EXAMPLE

In this section, we consider a repair problem in a storage network (partly shown in Fig. 1) as an example and explore efficient ways of solving it via scalar linear codes. Repair problems arise in distributed storage systems and are of practical interest. In this scenario, sources initially communicate with the middle nodes (encoders) R_i 's, and become inactive afterwards. The goal is then to be able to reconstruct the source messages by accessing only the encoders, in particular recover X_1, X_2 from messages of every pair of V_i 's at nodes W_j (W_1, \dots, W_6) and moreover exactly reconstruct each encoder message (output of R_i) from the other 3 encoders' messages at nodes U_1, \dots, U_4 (in this sense this is an "exact repair" type problem [14]). One question is then to determine the minimum amount of information that needs to be communicated between V_i 's and U_i 's (or in other words the minimum link capacity c) to achieve this goal. All other links are assumed to be unit capacity. This network is motivated in [14] where they show that the optimal link capacity $c = 0.5$ is achieved by using a two dimensional vector linear code with operations over a relatively large finite field. Here we obtain the minimum c and an achievability scheme when using only scalar linear codes.

Let X_1, X_2, X_3, X_4 , represent random variables of the incoming edges of V_1, \dots, V_4 , and X_5, \dots, X_{16} be the variables of edges with capacity c . Note that edges connecting V_i 's and W_j 's do not introduce new variables since they are unit capacity and V_i sends its incoming message X_i to W_j . The network topology enforces,

$$h(X_1, X_2) = h(X_1) + h(X_2), \quad (9)$$

$$h(X_{in}(T) \cup X_{out}(T)) - h(X_{in}(T)) = 0, \quad (10)$$

where T represents any node i.e. R_i, V_i, U_i and W_i and $X_{in}(T), X_{out}(T)$ denote the incoming and outgoing signals of node T respectively. Based on Theorem 4 one can find a scalar linear solution to a network by solving an LP over $\Phi_{n,2}^*$. However determining $\Phi_{n,2}^*$ in an efficient way becomes tricky when n grows large. Fortunately constraints and symmetries of networks usually allow us to skip the characterization of the whole $\Phi_{n,2}^*$ for large n and solve it more efficiently. For one thing note that when solving a network, one is only interested in those entries of the entropic vector that appear in network constraints and therefore only the projection of $\Phi_{n,2}^*$ onto these entries matters. As an example only about 40 elements of the $2^{16} - 1$ dimensional entropy vector of Fig. 1 show up in the constraints. Denote the indices of these elements by S . To refrain from characterizing $\Phi_{n,2}^*$ (here $\Phi_{16,2}^*$), we propose an alternative way of characterizing only feasible S -entry subsets.

Definition 4. Assume that v is a $2^n - 1$ dimensional vector whose entries are indexed by subsets of $\{1, \dots, n\}$. Let $K \subseteq \{1, \dots, n\}$ be defined as, $K = \{i | v_i = 1\}$. Then for $i, j \in \{1, \dots, n\}$ we define the binary relation \sim as follows:

$$i \sim j \text{ iff } \exists T \subset \{1, 2, \dots, n\} : i, j \in T \cap K \text{ and } v_T = 1. \quad (11)$$

Now we state the main theorem of this section, proof of which is omitted for brevity.

Theorem 5. Let $v \in \{0, 1, 2\}^{2^n - 1}$ be a $2^n - 1$ dimensional vector whose elements satisfy the submodularity conditions and $K \subseteq \{1, \dots, n\}$ is such that $\forall i \in K, v_i = 1$. The binary relation \sim is an equivalence relation on the set K for v iff v is a rank vector corresponding to an $n \times 2$ matrix.

Next, we use this theorem to find an algorithm to construct the region corresponding to a subset of entries of a rank vector.

Corollary 1. Let \hat{v} be $|S|$ dimensional sub-vector of a $2^n - 1$ dimensional vector s.t. $\hat{v} \in \{0, 1, 2\}^{|S|}$. \hat{v} corresponds to an $|S|$ -entry sub-vector of a rank vector of an $n \times 2$ matrix, iff:

- 1) it satisfies the submodularity conditions.
- 2) the relation \sim partially defined according to (11) on \hat{v} , satisfies the transitivity property: $a \sim b, b \sim c \Rightarrow a \sim c$.

Corollary 1 simplifies finding the region induced by some specific entries of the rank vector a lot when $|S|$, i.e. the number of entries of interest, is relatively small. Note that when we are looking at networks this is generally the case. However, in our example, if we are interested in a solution obtained by timesharing, we can simplify even more. Let V_i^- be all the V_j 's except V_i , i.e., $V_i^- = \{V_1, V_2, V_3, V_4\} \setminus V_i$. Let $X_{4+i}, X_{8+i}, X_{12+i}$ correspond to the three outgoing edges of V_i^- used to construct X_i . Then for any solution we have,

$$h(X_1) = h(X_2) = h(X_3) = h(X_4) = 1 \quad (12)$$

$$h(X_1, X_2) = h(X_1, X_3) = \dots = h(X_3, X_4) = 2 \quad (13)$$

$$h(X_1, X_2, X_3) = h(X_1, X_2, X_4) = 2 \quad (14)$$

$$h(X_{4+i}, X_{8+i}, X_{12+i}) = h(X_i, X_{4+i}, X_{8+i}, X_{12+i}) \quad (15)$$

$$h(X_{4+i}, X_{8+i}, X_{12+i}) \geq 1 \quad \forall i \in \{1, 2, 3, 4\} \quad (16)$$

$$h(X_1, X_6) = 1 \quad \text{and 11 similar equations.} \quad (17)$$

Considering these conditions the only unknown entries of the sub-vector h are $h(X_5), \dots, h(X_{16}), h(X_{4+i}, X_{8+i}, X_{12+i})$ for all $i \in \{1, \dots, 4\}$. Moreover since $c \leq 1$ we get the following,

Lemma 4. Any assignment for the entries of h that satisfies the following two conditions corresponds to a valid rank vector:

- h satisfies Equations (12-17)
- At most one of $h(X_{4+i}), h(X_{8+i}), h(X_{12+i})$ is zero

Using this lemma we get 4098 vectors. The next lemma is about the convex cone of these vectors.

Lemma 5. All of these 4098 vectors are boundary rays.

Let $\hat{\Phi}$ be the convex hull of these 4098 vectors. We can write the following LP to find c ,

$$\begin{aligned} \min \quad & c \\ \text{s.t.} \quad & h \in \hat{\Phi} \\ & h(X_i) < c \quad \forall i \in \{5, 6, \dots, 16\} \end{aligned} \quad (18)$$

Note that the other constraints of the network are captured in the region $\hat{\Phi}$ in LP (18). By solving the above LP, we get $c^* = \frac{2}{3}$. This is achieved by an equal timesharing of all rank vectors in $\hat{\Phi}$ that have exactly one zero element among $h(X_{4+i}), h(X_{8+i}), h(X_{12+i})$ for $i = 1, 2, 3, 4$. Although as stated earlier, a vector linear code for this network gives $c = 0.5$ [14], analyzing networks using vector linear codes is generally much more cumbersome. Therefore despite some capacity loss, scalar linear codes are appealing in practice where simple implementation is of crucial importance.

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Tran. on Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [2] Xijin Yan, Raymond W. Yeung, and Zhen Zhang, "The capacity region for multi-source multi-sink network coding," in *IEEE Int'l. Symp. Information Theory (ISIT)*, 2007.
- [3] B. Hassibi and S. Shadbakht, "Normalized entropy vectors, network information theory and convex optimization," in *Information theory workshop, Bergen, Norway*, 2007.
- [4] Z. Zhang and R. Yeung, "On characterization of entropy function via information inequalities," *IEEE Trans. on Information Theory*, vol. 44, no. 4, pp. 1440–1452, 1998.
- [5] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Tran. on Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, 2005.
- [6] S.-Y.R. Li, R.W. Yeung, and N. Cai, "Linear network coding," *IEEE Tran. on Inf. Theory*, vol. 49, pp. 371–381, February 2003.
- [7] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Tran. on Networking*, vol. 11, no. 5, pp. 782–795, 2003.
- [8] T. H. Chan, "Group characterizable entropy functions," in *IEEE Int. Symp. on Inf. Theory (ISIT)*, 2007, pp. 506–510.
- [9] R.W. Yeung, S.-Y.R. Li, N. Cai, and Z. Zhang, *Network coding theory*, now Publishers Inc, 2006.
- [10] A. W. Ingleton, "Representation of matroids," in *Combinatorial mathematics and its applications*, D. Welsh, Ed. London: Academic Press, 1971, pp. 149–167.
- [11] D. Hammer, A. Romashchenko, A. Shen, and N. Vereshchagin, "Inequalities for Shannon entropy and Kolmogorov complexity," *Journal of Computer and System Sciences*, , no. 60, pp. 442–464.
- [12] S. Shadbakht and B. Hassibi, "manuscript," .
- [13] D. Fong, S. Shadbakht, and B. Hassibi, "On the entropy region and the Ingleton inequality," in *MTNS*, 2008.
- [14] A. G. Dimakis, P. B. Godfrey, M. Wainwright Y. Wu, and K. Ramchandran, "Network coding for distributed storage systems," *submitted to IEEE Trans. on Inf. Theory*.